OpenAMのSAML利用時の認証方式の指定について

OSSTech

オープンソース・ソリューション・テクノロジ株式会社
作成日：2015/04/17
相本 智仁

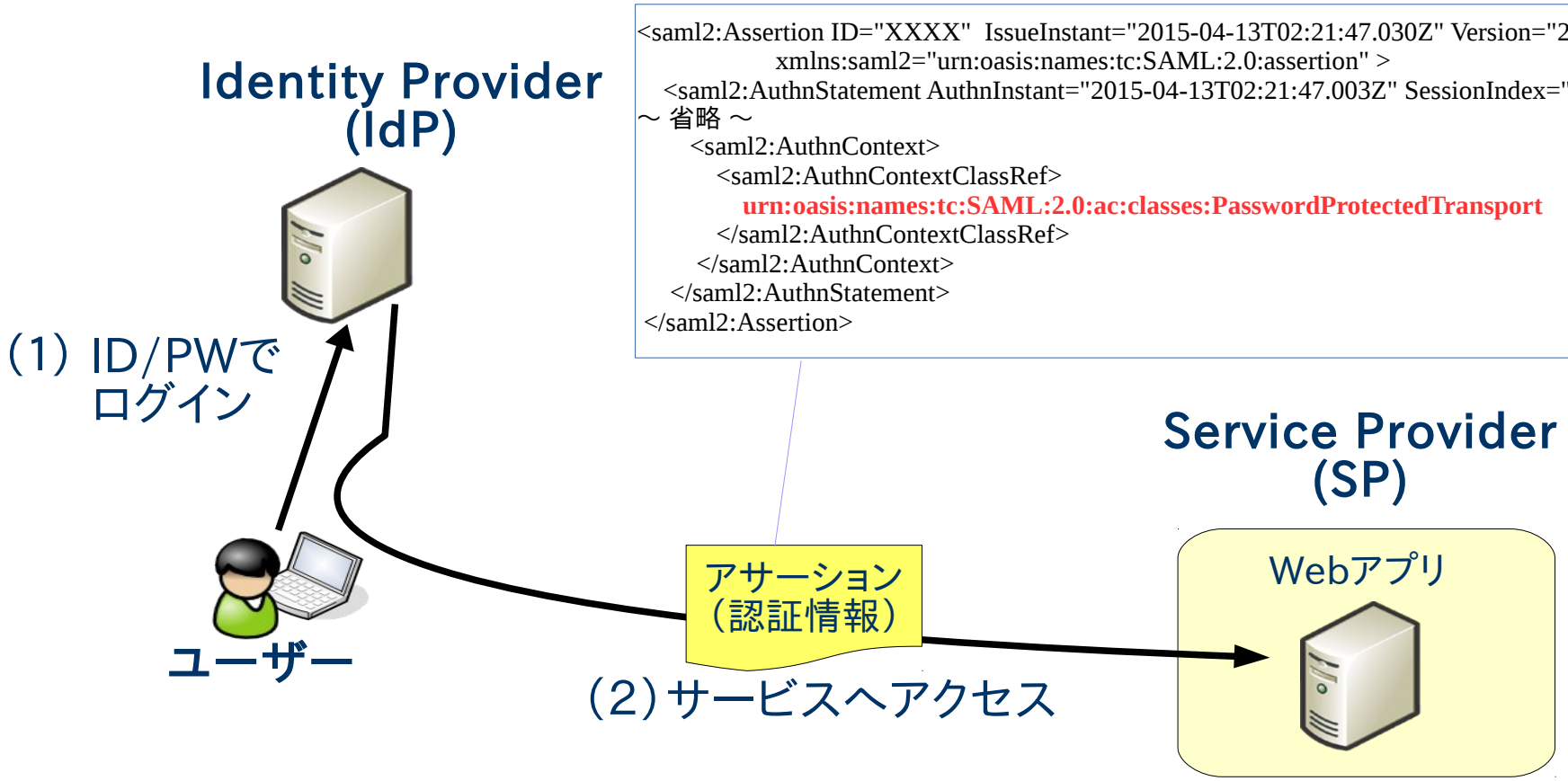
はじめに

- 本資料の目的
  - SAMLの認証方式の指定について把握する
    - Service Provider(SP)側の視点
      - 利用するIdPに認証方式を指定
    - Identity Provider(IdP)側の視点
      - SPに実施した認証方式の応答
  - OpenAMの実装を把握する
    - OpenAMをSPとして構築した場合
    - OpenAMをIdPとして構築した場合
      - どんな設定があり何ができるのか

SAMLの仕様
(認証コンテキストクラスについて)

AuthnContextとは

- SAML IdP が SPに応答するアサーションに含める認証コンテキストの情報(認証の情報)
- SAMLアサーションには必須(MUST)の要素




```
<saml2:Assertion ID="XXXX"  IssueInstant="2015-04-13T02:21:47.030Z" Version="2.0"
                xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" >
   <saml2:AuthnStatement AuthnInstant="2015-04-13T02:21:47.003Z" SessionIndex="XXXX">
～ 省略 ～
      <saml2:AuthnContext>
         <saml2:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
         </saml2:AuthnContextClassRef>
      </saml2:AuthnContext>
   </saml2:AuthnStatement>
</saml2:Assertion>
```

authnContextClassRef

- 認証コンテキストクラス
  - ユーザーの認証方式が示される
  - saml-authn-context-2.0(※)で規定されている

※ http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf

- <AuthnContext>の<AuthnContextClassRef>によりアサーションを受け取ったSPは、IdPでどのような認証が行われたか判断できる。

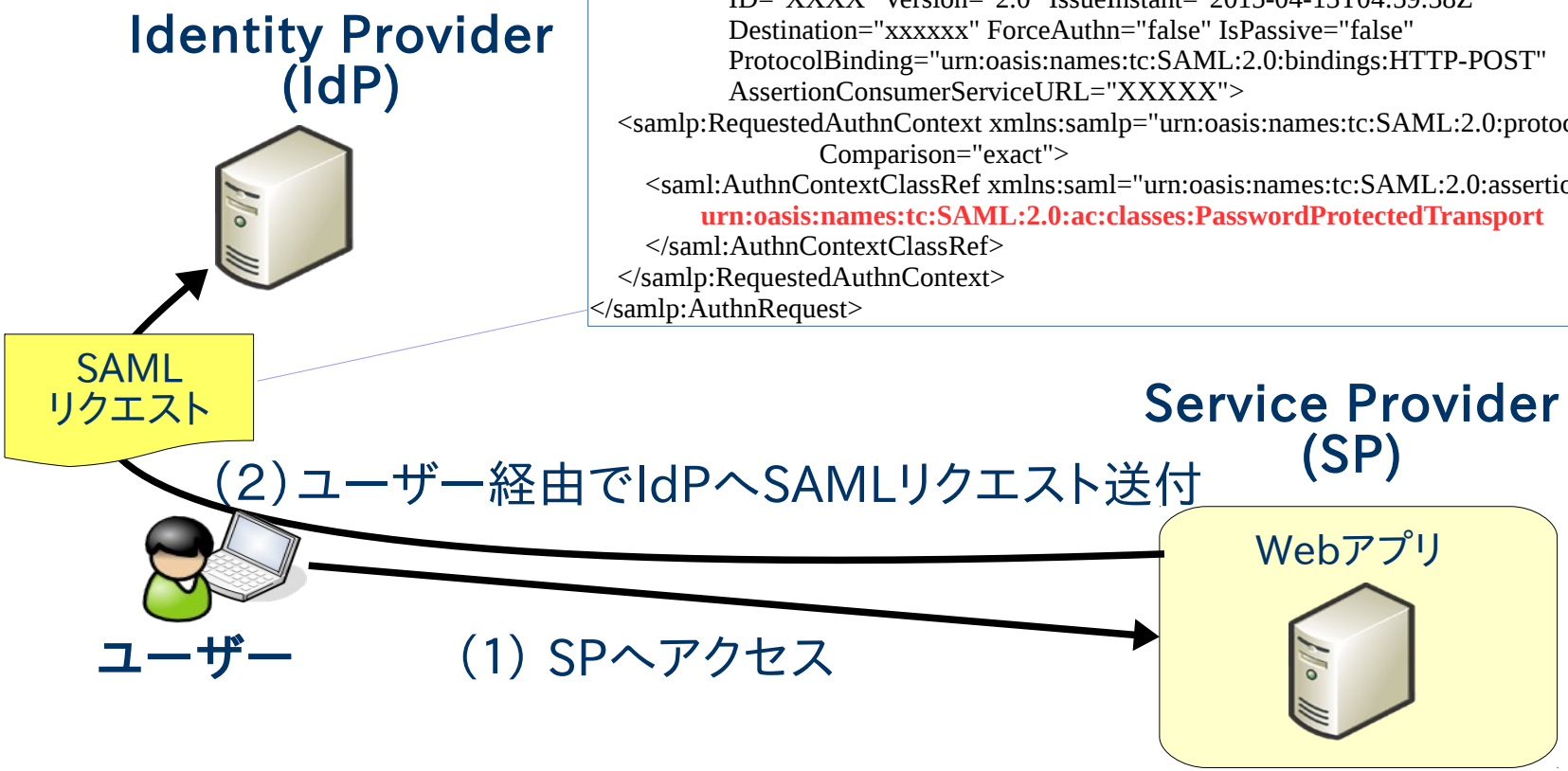
表：認証コンテキストクラスの例

認証コンテキストクラス	説明
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport	パスワードを提示して認証したことを示す
urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession	セッション等を使用し、すでに認証済みであること示す。過去のある時点で認証したこと示す。
urn:oasis:names:tc:SAML:2.0:ac:classes:X509	デジタル署名により認証したことを示す。
urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified	不特定の方法で認証したことを示す。

SPからの認証コンテキストクラスの指定

- 認証コンテキストクラスはSAMLリクエストに含めることができる。
  - SPはIdPに対して認証コンテキストを指定できる
  - RequestedAuthnContext要素で指定する

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                    ID="XXXX" Version="2.0" IssueInstant="2015-04-13T04:59:38Z"
                    Destination="xxxxxx" ForceAuthn="false" IsPassive="false"
                    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                    AssertionConsumerServiceURL="XXXXX">
  <samlp:RequestedAuthnContext xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                               Comparison="exact">
    <saml:AuthnContextClassRef xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
      urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
    </saml:AuthnContextClassRef>
  </samlp:RequestedAuthnContext>
</samlp:AuthnRequest>
```



RequestedAuthnContextの構成

- <AuthnContextClassRef>要素
  - 認証コンテキストクラスを一つ以上記述
- Comparison属性
  - 認証コンテキストクラスの比較手法を指定
  - 属性値としてexact、minimum、maximum、betterのどれか
  - デフォルト(省略された場合)はexactとなる
    - exactは指定された認証コンテキストの少なくとも一つには完全に一致していなければならい(MUST)

■RequestedAuthnContextの例

```
<samlp:RequestedAuthnContext xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                             Comparison="exact">
  <saml:AuthnContextClassRef xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
  </saml:AuthnContextClassRef>
</samlp:RequestedAuthnContext>
```

RequestedAuthnContextの構成

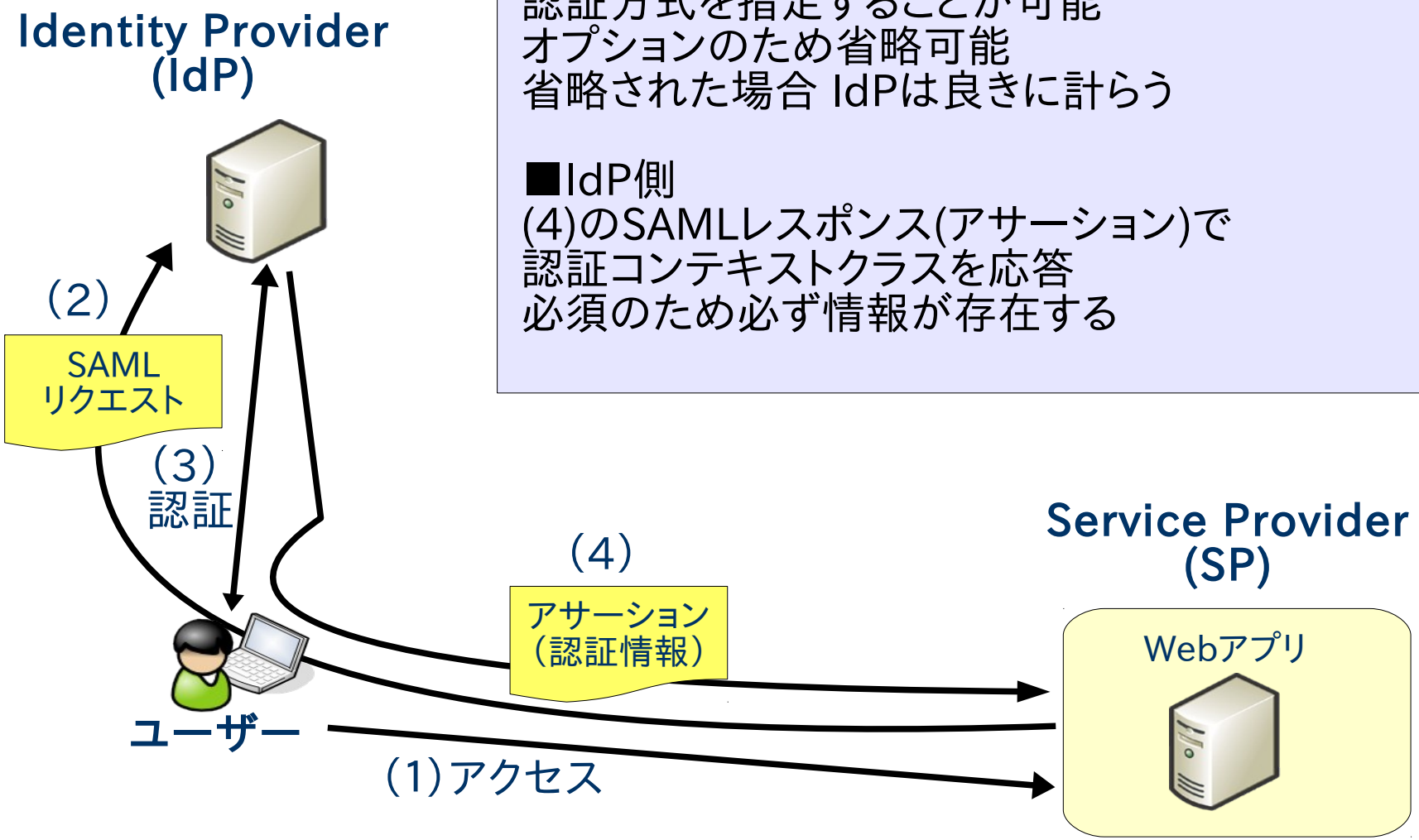
- RequestedAuthnContexは省略可能
  - 多くのSPでは使用していないようだ
    - Google や SalesforceのSAML SP が発行するSAMLリクエストを見るとRequestedAuthnContextは無い。
    - Shibboleht SPもデフォルト設定ではSAMLリクエストにRequestedAuthnContextは無い。
  - OpenAM はデフォルトで使用している
    - OpenAMをSPとして構築するとデフォルトとしてurn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransportをセットする設定となる。
- [参考]GoogleのSAMLリクエスト

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                    ID="pjafhmipjmejfnniheapofilmeheppjgijkloeni"
                    Version="2.0"
                    IssueInstant="2015-04-14T03:16:05Z"
                    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                    ProviderName="google.com"
                    IsPassive="false"
                    AssertionConsumerServiceURL="https://www.google.com/a/test.osstech.example.jp/acs">
  <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">google.com/a/test.osstech.example.jp</saml:Issuer>
  <samlp:NameIDPolicy AllowCreate="true"
                      Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
                      />
</samlp:AuthnRequest>
```

ここまでのまとめ

■SP側
(2)のSAMLリクエスト時に認証コンテキストクラスで
認証方式を指定することが可能
オプションのため省略可能
省略された場合 IdPは良きに計らう

■IdP側
(4)のSAMLレスポンス(アサーション)で
認証コンテキストクラスを応答
必須のため必ず情報が存在する

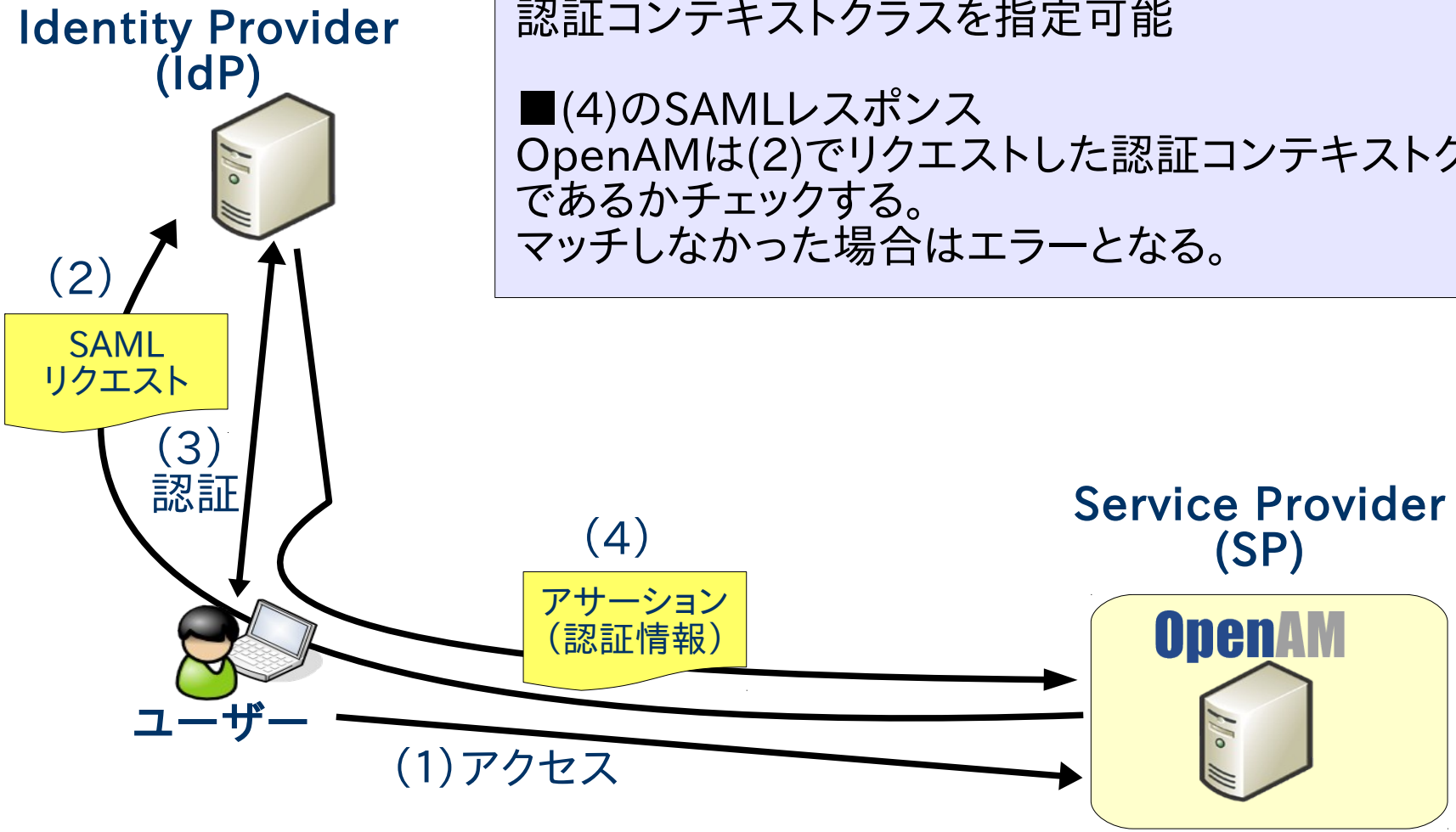


OpenAMの動作
(OpenAMをSPとして構築)

OpenAM SPとした場合

■(2)のSAMLリクエスト
「(1)アクセス」の際にクエリパラメータで
認証コンテキストクラスを指定可能

■(4)のSAMLレスポンス
OpenAMは(2)でリクエストした認証コンテキストクラス
であるかチェックする。
マッチしなかった場合はエラーとなる。



SAMLリクエスト時(SPイニシエイト)

- spSSOInit.jsp にクエリパラメータで指定可能
  - IdPに対してクエリパラメータで指定した認証コンテキストクラスをセットしたSAMLリクエストを送付可能
    - リクエストのURL例

```
https://sso.osstech.example.co.jp/openam/saml2/jsp/spSSOInit.jsp?
idpEntityID=[IdPのエンティティID]&metaAlias=/sp&
AuthnContextClassRef=urn:oasis:names:tc:SAML:2.0:ac:classes:X509
```

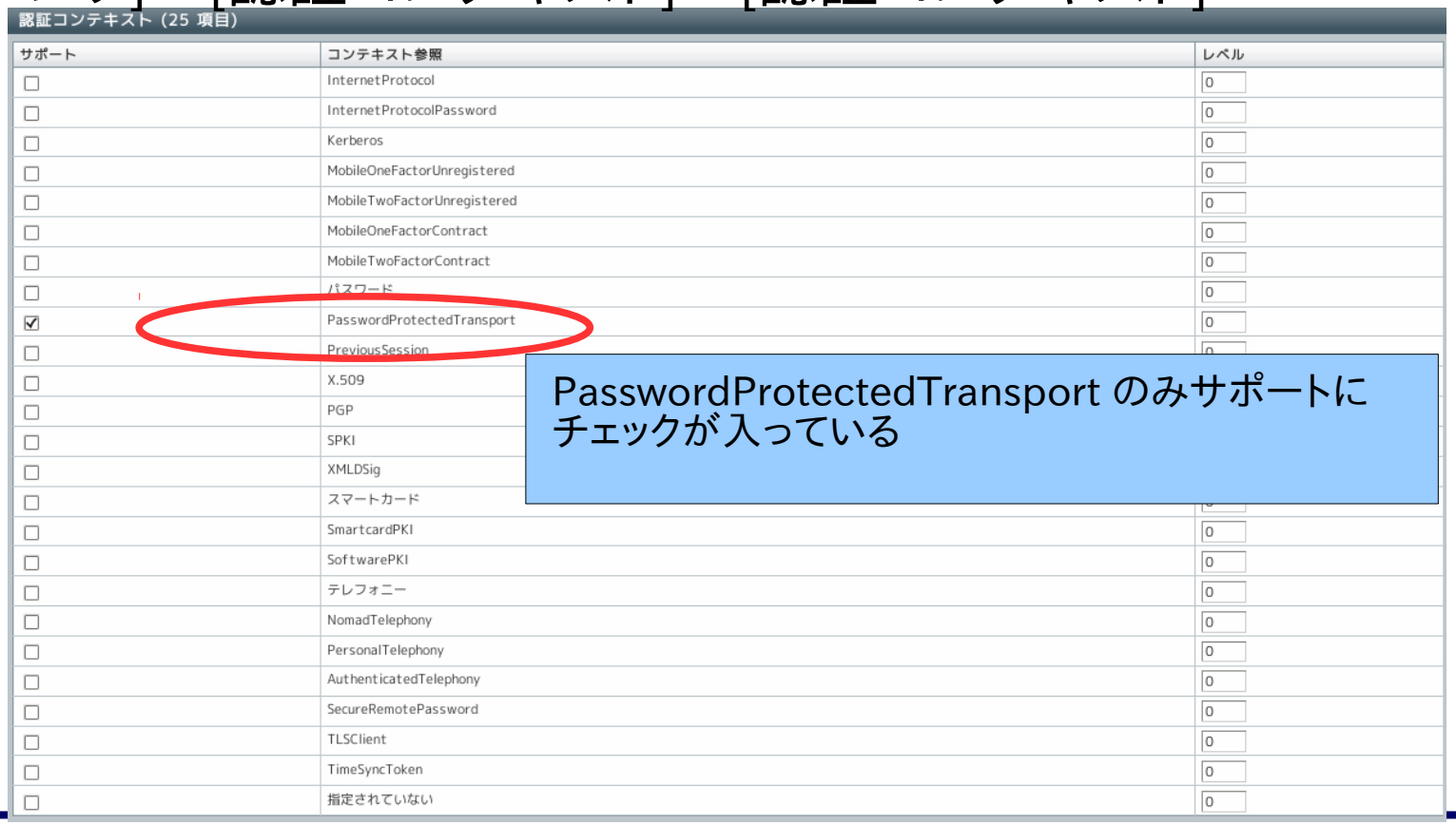
  - |(パイプ)で区切ることで複数の<AuthnContextClassRef>を指定することが可能
    - リクエストのURL例

```
https://sso.osstech.example.co.jp/openam/saml2/jsp/spSSOInit.jsp?
idpEntityID=[IdPのエンティティID]&metaAlias=/sp&
AuthnContextClassRef=urn:oasis:names:tc:SAML:2.0:ac:classes:X509|
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport|
urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified
```

複数指定可能だが、デフォルトの設定では
PasswordProtectedTransportのみ使用可能

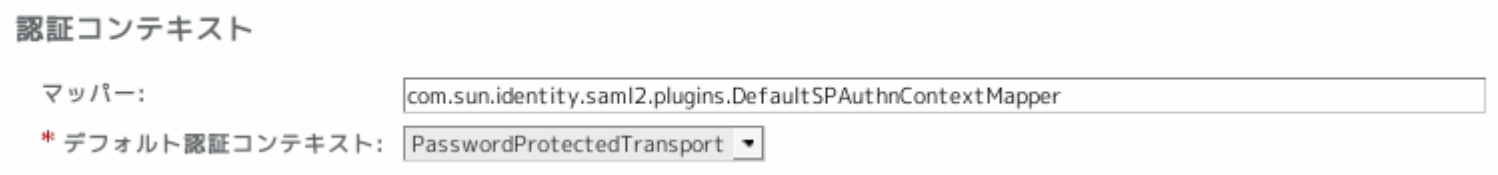
SPとして許容する認証コンテキスト1

- デフォルト設定では urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport のみ許容
  - [連携]→[ホストSPのEntityID(OpenAM自身)]→[表明コンテンツ]→[認証コンテキスト]→[認証コンテキスト]



SPとして許容する認証コンテキスト2

- クエリ指定がなければデフォルト設定のコンテキスト
  - [連携]→[ホストSPのEntityID(OpenAM自身)]→[表明コンテンツ]→[認証コンテキスト]→[デフォルトの認証コンテキスト]



  - OpenAMをSPとした場合、必ずSAMLリクエストに認証コンテキストが入る
    - ソースコードを見ると設定が見つからなくてもPasswordProtectedTransportをセットするよう書かれているため必ずセットされるようだ。

■DefaultSPAuthnContextMapper.javaより

```
// if list empty set the default
if (authCtxList.isEmpty()) {
    authCtxList.add(
        SAML2Constants.CLASSREF_PASSWORD_PROTECTED_TRANSPORT);
}
```

    - これはDefaultSPAuthnContextMapper.javaを使用するからであり、SPAuthnContextMapperを自作してしまえば自由に変えられる。

SAMLアサーションのチェック

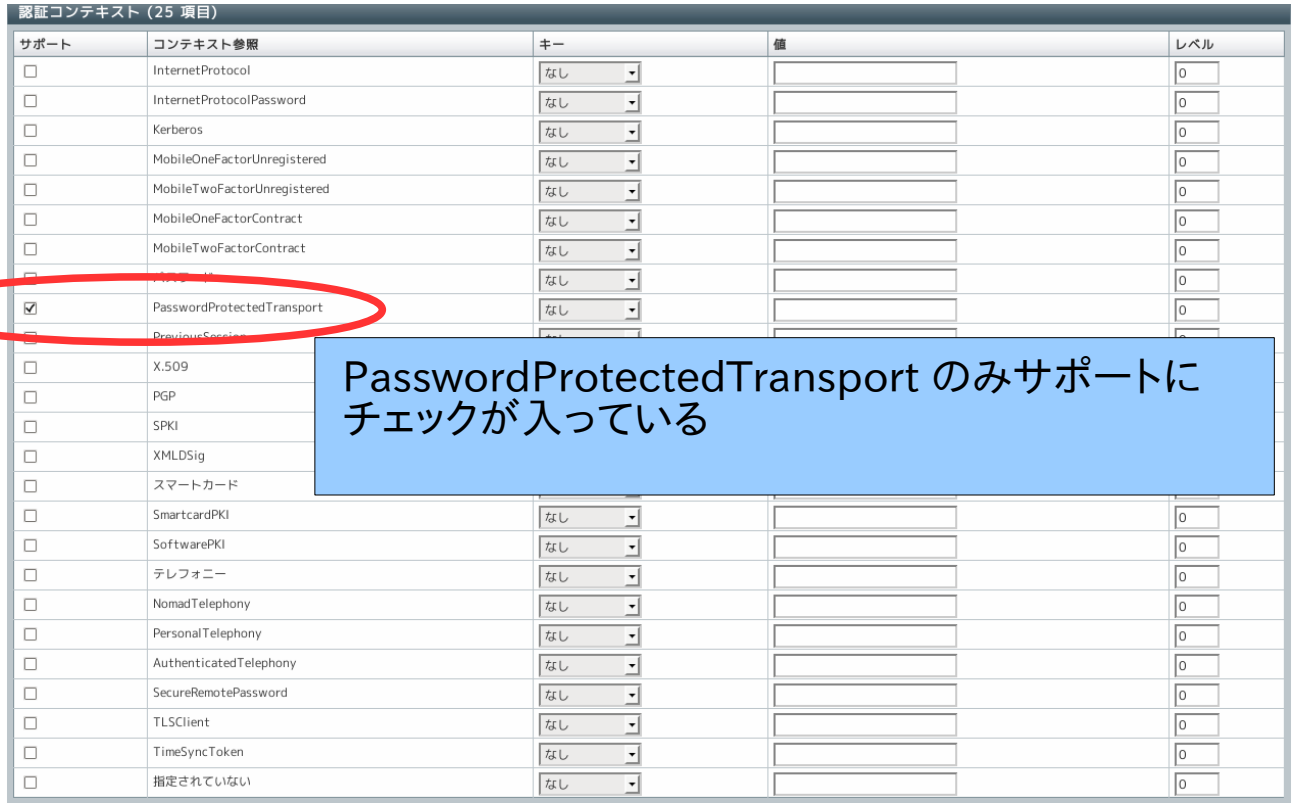
- IdPから送られてきたSAMLアサーションの認証コンテキストクラスはチェックを行う
  - リクエスト時に指定した認証コンテキストクラスと比較する
  - SAMLリクエストと異なった認証コンテキストで応答があった場合はSSO faildとする。(500のエラー画面応答)
  - 許容しない認証コンテキストクラスの場合も同様のエラー(SSO faild)
    - SPinitiatedのSAMLリクエスト生成時はエラーにならない。。。
    - IdPからのアサーションチェック時にエラーとなる
  - DebugログのFederationに以下のログあり

```
(osstech-openam11で発生させた場合)
libSAML2:04/13/2015 06:25:17:598 PM JST: Thread[ajp-bio-8009-exec-10,5,main]
ERROR: spAssertionConsumer.jsp: SSO failed.
com.sun.identity.saml2.common.SAML2Exception: AuthnContext doesn't match RequestedAuthnContext.
        at com.sun.identity.saml2.plugins.DefaultSPAuthnContextMapper.getAuthLevel(DefaultSPAuthnContextMapper.java:339)
        at com.sun.identity.saml2.common.SAML2Utils.fillMap(SAML2Utils.java:910)
        at com.sun.identity.saml2.common.SAML2Utils.verifyResponse(SAML2Utils.java:648)
        at com.sun.identity.saml2.profile.SPACSUtils.processResponse(SPACSUtils.java:1053)
        at org.apache.jsp.saml2.jsp.spAssertionConsumer_jsp._jspService(spAssertionConsumer_jsp.java:233)
```

OpenAMの動作
(OpenAMをIdPとして構築)

IdPとして許容する認証コンテキスト1

- デフォルト設定では urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport のみ許容
  - [連携]→[ホストIdPのEntityID(OpenAM自身)]→[表明コンテンツ]→[認証コンテキスト]→[認証コンテキスト]



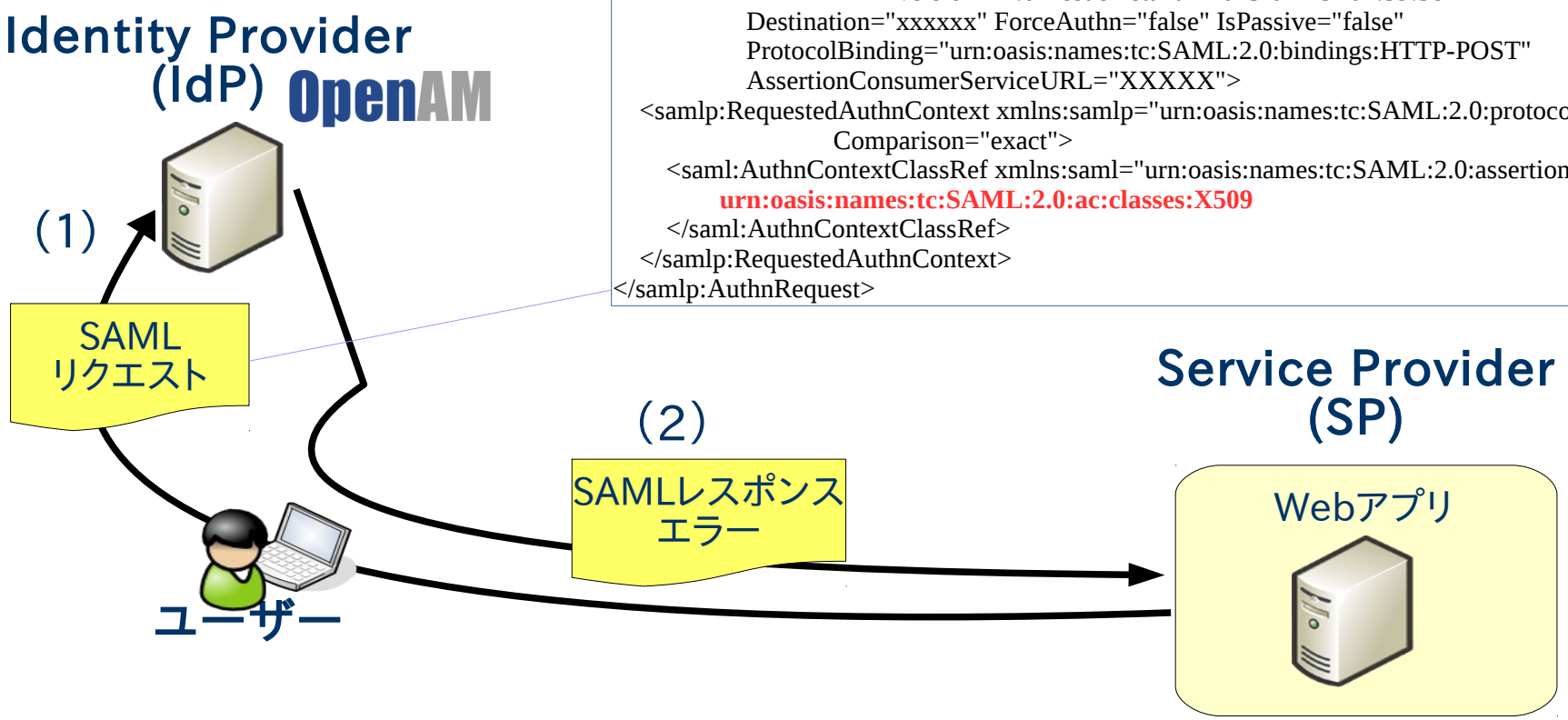
IdPとして許容する認証コンテキスト2

- SPからのSAMLリクエストでRequestedAuthnContexが省略された場合（SPから認証コンテキストクラスの指定が無い場合）
  - 設定による。デフォルトでは urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransportと設定されている
  - [連携]→[ホストIdPのEntityID(OpenAM自身)]→[表明コンテンツ]→[認証コンテキスト]→[デフォルトの認証コンテキスト]

サポートしない認証コンテキスト1

- SAMLリクエストがサポートしない認証コンテキスト
  - 例えばPasswordProtectedTransport以外のSAMLリクエストがOpenAMに届いたら
  - OpenAMはSPにエラーのSAMLレスポンスを応答する

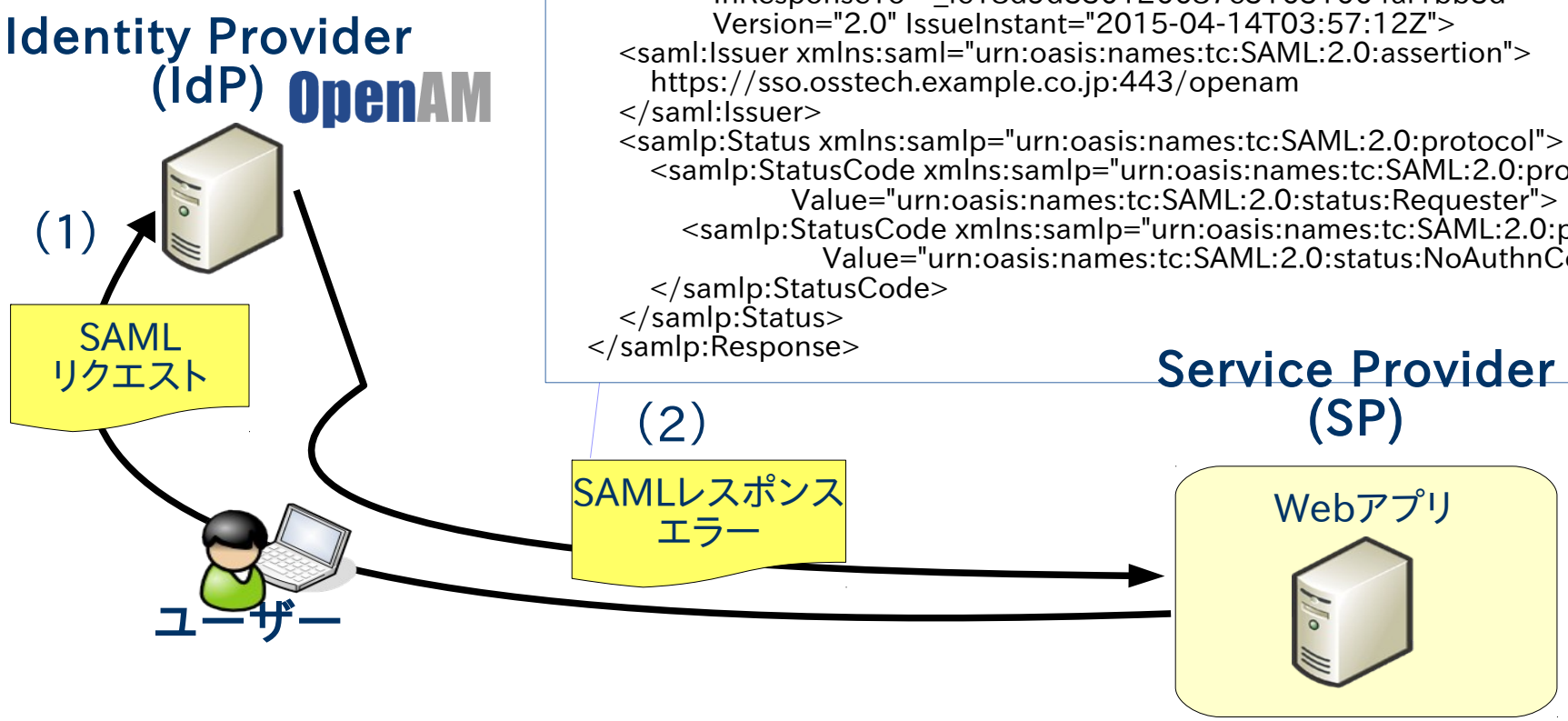
```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                    ID="XXXX" Version="2.0" IssueInstant="2015-04-13T04:59:38Z"
                    Destination="xxxxxx" ForceAuthn="false" IsPassive="false"
                    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                    AssertionConsumerServiceURL="XXXXX">
  <samlp:RequestedAuthnContext xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                               Comparison="exact">
    <saml:AuthnContextClassRef xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
      urn:oasis:names:tc:SAML:2.0:ac:classes:X509
    </saml:AuthnContextClassRef>
  </samlp:RequestedAuthnContext>
</samlp:AuthnRequest>
```



サポートしない認証コンテキスト2

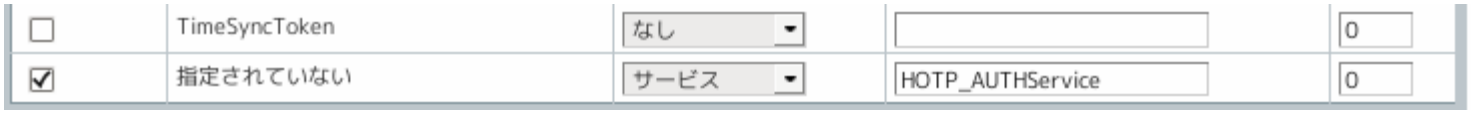
- OpenAMではDebugレベルでエラーはなし
  - SAML2.accessにSAML応答が出力(SAML2.errorは無し)
    - SAMLとしてはエラーだけどOpenAMとしてはエラーではないためエラーログにはでないと思われる。

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                ID="s2852881855bf62d1aa4b040d3e134e6ded57d70e1"
                InResponseTo="_fc18d9d530126687c51651664af1bb5d"
                Version="2.0" IssueInstant="2015-04-14T03:57:12Z">
   <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
      https://sso.osstech.example.co.jp:443/openam
   </saml:Issuer>
   <samlp:Status xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
      <samlp:StatusCode xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                        Value="urn:oasis:names:tc:SAML:2.0:status:Requester">
         <samlp:StatusCode xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                           Value="urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext"/>
      </samlp:StatusCode>
   </samlp:Status>
</samlp:Response>
```



認証コンテキストとOpenAMの認証1

- 認証コンテキストとOpenAMの認証を紐づけられる
  - デフォルトは指定がないため、/openam/UI/Login?realm=[レルム名]へforwardした認証が行われる
- 認証コンテキスト毎方式を変えられる
  - モジュール単位の認証(module=xxx)
  - 認証連鎖(service=xxx)
  - 認証レベル(authlevel=xxx)　・・・等々
- 設定例：IDPの認証コンテキスト



この設定を行うと、SAMLリクエストが
urn:oasis:names:tc:SAML:2.0:ac:classes:unspecifiedだった場合は
/openam/UI/Login?realm=[レルム名]&service=HOTP_AUTHSERVICE とアクセスした際の認証を行う

※urn:oasis:names:tc:SAML:2.0:ac:classes:unspecifiedのことを「指定されていない」という日本語訳は判りづらい…

認証コンテキストとOpenAMの認証2

- 認証コンテキストのチェック等を行うソースコードはIDPSSOFederate.javaのdoSSOFederate()
- 認証を行うURLは同ソース内のredirectAuthentication()あたりを見る
  - ここでIdPの設定からservice=xxやmodule=xxをセットする。
  - SP単位の設定は無さそう
    - 認証コンテキスト単位で認証方式は決めている
    - 例えば、SP1から認証コンテキストクラスの指定が無い場合、OpenAMでこのSP1だけ特定の認証コンテキストクラスを強制的に使用させたいということは標準ではできない
      - プラグイン(アダプタ)を開発すればいいのかな…

まとめ

- OpenAMをSPとした場合
  - 認証コンテキストクラスは必ずセットされる
  - spSSOInit.jspはクエリパラメータで色々指定でき自由度が高い
  - その反面クエリパラメータはユーザーが操作できてしまう
    - spSSOInit.jspは直接アクセス出来ないようにして、forwardする等してサーバー側で制御して活用すれば用途に沿ったSAMLリクエストが発行できると思われる。
- OpenAMをIdPとした場合
  - 認証連鎖、認証モジュール、認証レベルと用途に沿った指定が、自由度高く色んな要件にマッチしそう。
  - とはいえ実際は多くのSAMLのSPが認証コンテキストクラスを指定して来ないので、あまり使うことは無さそう